

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Penelitian ini menggunakan beberapa sumber pustaka yang berhubungan dengan kasus atau metode yang akan diteliti, diantaranya adalah sebagai berikut:

I Gede Handika (2018) “Memanfaatkan *Framework Laravel* dalam pembangunan aplikasi *E-Travel* berbasis website” . yang bertujuan untuk membangun situs web travel yang dapat membantu konsumen dalam melakukan pemesanan tiket secara online. Dan penelitian ini menghasilkan pemesanan tiket pesawat, kereta, hotel. Restaurant dan tiket atraksi/event.

Rifky Lana Rahardian, Nih Luh Gede Pivin Suwirmayanti (2020) dalam penelitiannya “mengimplementasikan framework laravel dalam *E-Tourism* Provinsi Bali berbasis Web”. Penelitian ini menghasilkan pemesanan tiket pesawat, kereta, hotel, Restaurant dan tiket atraksi/event. Sehingga dapat memberikan informasi secara luas berupa gambar, tulisan , dan pemetaan lokasi.

Irma Yunita Ruhiawati, Siti Sopiah (2019) dalam penelitiannya “Membangun sistem informasi akademik sekolah berbasis web dengan laravel”. Pada penelitian ini menghasilkan pusat informasi yang lebih efisien dan dapat diakses oleh siswa dengan mudah dan cepat.

Benhur berahmana (2020) dalam penelitiannya “mengimplementasikan framework laravel pada sistem informasi pariwisata kabupaten Karo berbasis web”. Dalam penelitian tersebut Benhur Berahmana membuat Website sebagai media

penghubung antara wisatawan dengan tempat wisata. Website tersebut akan berjalan pada browser yang memiliki koneksi internet. Dan, dengan Implementasi Framework Laravel tersebut maka memudahkan pengunjung web dapat melakukan pemesanan Tour Guide Pariwisata secara online.

Iis Nuraida (2021) dalam penelitiannya “Informasi Pariwisata di Kabupaten Pangandaran berbasis Website Menggunakan Teknologi *framework Laravel*”. Tujuan dari penelitian ini adalah menghasilkan aplikasi web yang berisi informasi untuk membantu pemakai/user dalam bidang penggunaan website khususnya web pariwisata. Dan dapat Mempermudah pengunjung mencari informasi Pariwisata di Kabupaten Pangandaran.

Sedangkan yang diusulkan dalam penelitian ini memiliki kesamaan dalam penggunaan framework laravel namun, pada penelitian ini lebih difokuskan kepada *Progressive Web Application(PWA)*, pusat informasi destinasi wisata di kabupaten Jepara dan ada rute tujuan di setiap destinasi wisata. Agar dapat memudahkan calon wisatawan yang akan berkunjung ke tempat Wisata tersebut.

Tabel 2. 1 Tinjauan Pustaka

Sumber	Objek	Masalah	Teknologi	Hasil
Rifky Lana Rahardian, Nih Luh Gede Pivin Suwirmayanti (2020)	E-Tourism Provinsi Bali berbasis Web dengan Framework Laravel	Bagaimana membangun e-tourism provinsi Bali berbasis web dengan framework Laravel	<i>Laravel,</i> <i>PHP,</i> <i>MySQL</i>	E-Tourism memberika n informasi secara luas berupa tulisan gambar dan

				pemetaan lokasi
Irma Yunita Ruhiawati, Siti Sopiah (2019)	Perancangan sistem informasi akademik sekolah berbasis web dengan laravel	Bagaimana membuat web sistem informasi sekolah menggunakan framework Laravel	<i>Laravel, PHP, MySQL</i>	Menghasilkan pusat informasi yang lebih efisien
Benhur berahmana (2021)	Merealisasikan sistem layanan informasi Pariwisata Kabupaten Karo berbasis website menggunakan Framework Laravel	Bagaimana merancang web informasi pariwisata yang dapat diakses dengan cepat dan mudah dimengerti dalam penyajiannya	<i>Laravel, MySQL</i>	Menghasilkan informasi pariwisata yang dapat diakses dengan cepat dan mudah

Iis Nuraida (2021)	Perancangan Dan Pembuatan Sistem Informasi Pariwisata Di Kabupaten Pangandaran Berbasis Website Menggunakan Teknologi Framework Laravel	Bagaimana Membangun sebuah aplikasi berbasis Web Yang Berupa Sebuah Sistem Informasi Tentang Pariwisata Yang Ada Di Pangandaran Yang dapat Di Akses Dengan Cepat	<i>Laravel,PHP,MySQL,Bootstrap</i>	Menghasilk an Aplikasi Web Yang Berisi Informasi Untuk Membantu Pemakai/U ser Dalam Bidang Penggunaa n Website Khususnya Web Pariwisata
I Gede Handika (2018)	Pemanfaatan Framework Laravel dalam Pembangunan <i>Aplikasi E- Travel</i> Berbasis Website	Bagaimana membangun aplikasi travel berbasis website dengan menggunaka n framework laravel	<i>Laravel,PHP,MySQL</i>	membangu n aplikasi travel yang dapat membantu proses pembelian tiket secara online
Usulan penelitian				
Makarno (2022)	Merealisasikan Sistem	Bagaimana merancang	<i>Laravel,MySQL</i>	Akan dibuat

	Informasi Pariwisata Kabupaten Jepara Berbasis PWA Menggunakan Framework Laravel	PWA informasi pariwisata yang dapat di akses dengan cepat dan mudah dimengerti dalam penyajianya		PWA informasi pariwisata menggunakan an framework laravel
--	---	---	--	---

## 2.2 Dasar Teori

### 2.2.1 Framework Laravel

Menurut Siena, (2009) *Framework* adalah sekumpulan library yang diorganisasikan pada sebuah rancangan arsitektur untuk memberikan kecepatan, ketepatan, kemudahan dan konsistensi di dalam pengembangan aplikasi dari definisi tersebut”. *Framework* terdiri dari:

#### 1. Model

Model mencakup semua proses yang terkait dengan pemanggilan struktur data baik berupa pemanggilan fungsi, input processing atau mencetak output ke dalam browser.

#### 2. View

View mencakup semua proses yang terkait layout output. Bisa dibayangkan untuk menaruh template interface website atau aplikasi.

### 3. Controller

Controller mencakup semua proses yang terkait dengan pemanggilan database dan kapsulisasi proses proses utama. Jadi semisal di bagian ini ada file bernamamember.php, maka semua proses yang terkait dengan member akan dikapsulisasi/ dikelompokan dalam file ini.

Beberapa alasan dari digunakannya framework dalam membuat aplikasi adalah sebagai berikut.

- a. Aplikasi akan memiliki standar pemrograman yang universal.
- b. Menghindari repetitive work.
- c. Memudahkan dalam team work.
- d. Memudahkan dalam maintenance dan pengembangan aplikasi dimasa mendatang.
- e. Hemat waktu dan biaya

Laravel merupakan Framework PHP yang menekankan pada kesederhanaan dan fleksibilitas pada desainnya. Laravel dirilis dibawah lisensi MIT dengan sumber kode yang disediakan di Github. Sama seperti framework PHP lainnya, Laravel dibangun dengan basis MVC (*Model-View-Controller*). Laravel dilengkapi command line tool yang bernama “Artisan” yang bisa digunakan untuk packaging bundle dan instalasi bundle. *Framework Laravel* dibuat oleh Taylor Otwell, proyek Laravel dimulai pada April 2011. Awal mula proyek ini dibuat karena Otwell sendiri tidak menemukan *framework* yang up- to-date dengan versi PHP. Mengembangkan *framework* yang sudah ada juga bukan merupakan ide yang bagus karena keterbatasan sumber daya. Dikarenakan beberapa keterbatasan tersebut,

Otwell membuat sendiri framework dengan nama Laravel. Oleh karena itu Laravel mensyaratkan PHP versi 5.3 keatas. (Rohman, 2014).

Framework Laravel juga memiliki beberapa keunggulan sebagai berikut:

- a. Menggunakan *Command Line Interface* (CLI) Artisan.
- b. Menggunakan package manager PHP Composer.
- c. Penulisan kode program lebih singkat, mudah dimengerti, dan ekspresif.

Kemudian untuk cara instalasi *framework Laravel* dapat dilakukan dengan 3 cara yaitu:

Melalui Installer Laravel.

- a. Menggunakan *Composer* dengan mengetikkan perintah *create-project*.
- b. Download source code Laravel secara lengkap melalui GitHub dengan alamat <https://github.com/laravel/laravel/>

Untuk menggunakan Laravel versi 5.3 atau di atasnya maka komputer atau server yang digunakan harus memenuhi persyaratan sebagai berikut:

- a. PHP  $\geq$  5.6.4
- b. OpenSSL PHP Extension
- c. PDO PHP Extension
- d. Mbstring PHP Extension
- e. Tokenizer PHP Extension
- f. XML PHP Extension

Fitur framework Laravel yang ditekankan pada penelitian ini adalah *Blade*, *Migration*, *Eloquent ORM*, *Authentication*, dan *Resource Controller*. Berikut adalah penjelasan mengenai lima fitur tersebut:

a. *Blade*

*Blade* adalah *template engine*. Pada dasarnya *Blade* adalah *view* namun dengan menggunakan *Blade* akan mempermudah untuk mengatur tampilan website dan menampilkan data. Cara untuk membuat file *view* menjadi file *Blade* adalah dengan menambahkan *ekstensi* `.blade.php` pada file *view*. Dan cara untuk memanggil file *Blade* sama dengan cara untuk memanggil file *view* biasa. Contoh program berikut adalah perbandingan antara file *view* biasa dengan file *Blade*.

b. *Migration*

*Migration* adalah fitur yang menyediakan cara baru untuk membuat *database*. Dengan menggunakan *migration* cara membuat *database* melalui *Command Line Interface (CLI) database* atau dengan menggunakan aplikasi *database manager* digantikan dengan menggunakan *class*. Tahapan menggunakan *migration* adalah membuat *class* kemudian melakukan perintah *migrate* melalui *Command Line Interface (CLI) artisan*.

Keuntungan menggunakan *migration* adalah *class* yang dibuat bisa dipakai untuk membuat *database* pada berbagai macam *Relational Database Management System (RDBMS)* yang didukung oleh *Laravel*. Sebagai contoh misalnya aplikasi yang digunakan selama ini menggunakan *database MySQL*, kemudian karena alasan pengembangan aplikasi maka akan dilakukan penggantian *database* ke *PostgreSQL*. Dalam proses penggantian tersebut tidak perlu membuat *class* lagi, tinggal melakukan perintah *migrate* melalui *Command Line Interface (CLI) artisan*.



Keuntungan lain dari menggunakan *migration* adalah semua perubahan yang dilakukan pada database akan disimpan pada suatu tabel. Sehingga bisa dilakukan pembatalan (*rollback*) pada database jika melakukan perubahan yang tidak benar.

c. *Eloquent ORM*

*Eloquent ORM* adalah implementasi dari *ActiveRecord* yang digunakan untuk mengatur relasi antar tabel di database. Pada *Eloquent ORM* tabel direpresentasikan dalam bentuk kelas dan data yang tersimpan didalam tabel direpresentasikan dalam bentuk objek. Relasi yang dapat diatur menggunakan *Eloquent ORM* adalah sebagai berikut:

- 1) *One-to-One* yaitu relasi satu ke satu. Pada relasi ini digunakan *method hasOne* dan *belongsTo*.
- 2) *One-to-Many* yaitu relasi satu ke banyak. Pada relasi ini digunakan *method hasMany* dan *belongsTo*.
- 3) *Many-to-One* yaitu relasi banyak ke satu. Pada relasi ini digunakan *method belongsTo* dan *hasMany*.
- 4) *Many-to-Many* yaitu relasi banyak ke banyak. Pada relasi ini digunakan *method belongsToMany*.

d. Resource Controller

*Resource Controller* adalah fitur yang digunakan untuk mempercepat pembuatan *controller*. Sebagai contoh misalnya ada *controller* yang menangani semua *HTTP request* terhadap data dosen, untuk membuat *controller* tersebut hanya perlu mengetikkan perintah berikut contoh.

```
php artisan make:controller DosenController --resource
```

Perintah tersebut akan menghasilkan controller `DosenController.php` yang disimpan pada folder `app/Http/Controllers`. Pada Tabel 2.2, ini adalah daftar *action* yang dapat dilakukan oleh controller `DosenController.php`.

Tabel 2. 2 Action DosenController

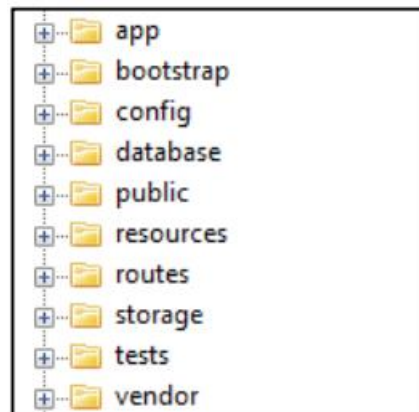
No	VERB	URI	Action	Route name
1	<i>GET</i>	/dosen	<i>Index</i>	Dosen.index
2	<i>GET</i>	/dosen/create	<i>Create</i>	Dosen.create
3	<i>POST</i>	/dosen	<i>Store</i>	Dosen.store
4	<i>GET</i>	/dosen/{dosen}	<i>Show</i>	Dosen.show
5	<i>GET</i>	/dosen/{dosen}/edit	<i>Edit</i>	Dosen.edit
6	<i>PUT/PATCH</i>	/dosen/{dosen}	<i>Update</i>	Dosen.edit
7	<i>DELETE</i>	/dosen/{dosen}	<i>Destroy</i>	Dosen.destroy

Pada Tabel 2.2 menunjukkan contoh daftar action pada `DosenController.php` yang terdiri dari *Verb*, *Uri*, *Action*, dan *Route Name*

Setelah membuat controller `DosenController.php` hal yang harus dilakukan selanjutnya adalah membuat satu baris kode program pada *route*.

```
Route::resource('dosen', 'DosenController');
```

Satu baris kode program pada route diatas akan menangani semua *route* untuk melihat, menambah, mengedit, dan menghapus data dosen. Jadi dapat disimpulkan dengan menggunakan fitur *Resource Controller* dapat mempercepat pembuatan *controller* serta dapat menyederhanakan route untuk controller. Struktur folder dari *framework Laravel 5.3* yang masih default dapat dilihat pada Gambar 2.2 Struktur Folder *Framework Laravel*



**Gambar 2. 1 Struktur Folder Framework Laravel**

Berikut adalah keterangan pada Gambar 2.2 Struktur Folder *Framework*

*Laravel*

- 1) *Folder app* adalah folder yang berisi kode program inti dari aplikasi yang akan dibuat. Model dan controller tersimpan pada folder ini.
- 2) *Folder bootstrap* adalah folder yang berisi konfigurasi *autoloading* dan terdapat juga *folder cache* yang menyimpan file-file yang dihasilkan secara otomatis oleh Laravel untuk mengoptimasi kinerja dari sistem yang dihasilkan.
- 3) *Folder config* adalah folder yang berisi semua file konfigurasi aplikasi.
- 4) *Folder database* adalah folder yang berisi file database *migration* dan *seeds*.
- 5) *Folder public* adalah folder yang berisi file *index.php*. File tersebut digunakan sebagai *entry point* untuk menangani semua *request* yang masuk ke aplikasi. Pada folder ini juga dapat disimpan beberapa aset dari aplikasi seperti gambar, JavaScript, dan CSS.
- 6) *Folder resources* adalah folder yang berisi file *view* dari aplikasi yang dibuat. Selain itu terdapat juga file language yang digunakan aplikasi.

- 7) *Folder routes* adalah folder yang berisi file yang digunakan untuk mendefinisikan semua route ke aplikasi. Secara default ada tiga file *route* yang disediakan Laravel yaitu *api.php*, *console.php*, dan *web.php*.
- 8) *Folder storage* adalah folder yang berisi template *Blade* yang dikompilasi, file *session*, file *cache*, dan file lainnya yang dihasilkan secara otomatis oleh Laravel.
- 9) *Folder test* adalah folder yang berisi semua file test yang dibuat untuk aplikasi.
- 10) *Folder vendor* adalah folder yang menyimpan semua *library* yang digunakan.

### 2.2.2 Progressive Web Apps

*Progressive Web Apps* (PWA) adalah konsep pengalaman pengguna yang menghubungkan bagian terbaik web dan bagian terbaik native apps. PWA berguna bagi pengguna sejak pertama membuka halaman sebuah web dengan konsep PWA, dan seiring dengan pengguna menggunakan aplikasi web lebih banyak lagi, aplikasi akan menjadi semakin powerful. Aplikasi dapat dimuat dengan cepat, bahkan dalam kondisi internet yang kurang baik, bisa mengirim push notifications, punya ikon aplikasi di home screen, dan bisa berjalan dalam mode layar penuh . Sebuah aplikasi PWA memiliki karakteristik sebagai berikut:

1. *Progressive*: berjalan dengan baik bagi semua pengguna, tidak memandang browser yang dipakai, karena dibangun dengan progressive enhancement dari sebuah inti aplikasi.
2. *Responsive*: ditampilkan dengan baik pada semua perangkat berbagai ukuran: desktop, tablet, mobile, dan apapun perangkat baru selanjutnya.

3. *Connectivity independent*: dengan *service worker* aplikasi dapat bekerja dalam kondisi offline atau jaringan yang lemah.
4. *App-like*: terasa seperti sebuah aplikasi, karena model *app shell* yang memisahkan fungsionalitas aplikasi dari kontennya.
5. *Fresh*: selalu *up-to-date* berkat pembaruan melalui *service worker*
6. *Safe*: aplikasi dilayani via HTTPS untuk mencegah snooping dan memastikan konten tidak diubah.
7. *Discoverable*: dapat diidentifikasi sebagai "*application*" karena manifest W3C dan *service worker registration scope*, dan memungkinkan mesin pencari untuk menemukannya.
8. *Re-engageable*: memudahkan dalam mengajak pengguna untuk menggunakan ulang aplikasi melalui fitur seperti push notifications.
9. *Installable*: memungkinkan pengguna untuk menambahkan aplikasi ke *homescreen* tanpa harus kerepotan dengan *app store*.
10. *Linkable*: dengan mudah dapat membagikan aplikasi dengan membagikan URL, dan tidak memerlukan instalasi yang rumit.

PWA sepenuhnya mengandalkan browser pengguna dan teknologi yang ada didalamnya. Sampai saat tugas akhir ini ditulis, sudah ada 73.61% dari seluruh browser di seluruh dunia yang mendukung fitur *service worker*, seperti Mozilla Firefox, Google Chrome, Chrome for Android dan Opera, sementara Edge dan Safari belum mendukung fitur ini. Ada tiga jenis arsitektur PWA yaitu *Server-Side Rendering (SSR)*, *Client-Side Rendering (CSR)*, dan gabungan keduanya. Arsitektur yang lazim digunakan sampai saat ini adalah SSR, dimana browser

mengambil data halaman melalui HTTP/HTTPS dan data dikembalikan berupa halaman lengkap dengan data dinamis sudah di-render. SSR bagus digunakan karena memberikan render pertama dengan cepat, tetapi memuat ulang sebuah halaman SSR berarti membuang seluruh data, dan akan memakan waktu ketika memuat kembali seluruh resource di halaman. SSR juga merupakan teknik yang sudah matang dengan banyak tool yang mendukung. SSR juga berjalan di semua browser tanpa harus memikirkan perbedaan *implementasi JavaScript* antar browser. Arsitektur CSR adalah ketika *JavaScript* berjalan di browser untuk memanipulasi DOM (Document Object Model).

Keuntungan dari CSR adalah memberikan umpan balik langsung ketika pengguna menekan tombol navigasi, ketimbang harus menunggu server untuk mengetahui informasi apa yang harus ditampilkan. CSR bisa memutuskan apakah harus memuat ulang seluruh halaman atau sebagian halaman saja ketika ada data baru yang masuk. Best practice yang disarankan adalah menggunakan gabungan keduanya, jadi render pertama dilakukan di *server* dan menggunakan data dari server. Lalu ketika klien menerima halaman tersebut, *service worker* menyimpan semua yang dibutuhkan untuk *app shell* ke dalam *cache*.

### 2.2.3 Mysql

Menurut Nugroho dalam buku Database Relasional dengan *MySQL* disebutkan bahwa *MySQL* adalah software sistem pengelolaan atau manajemen basis data yang berupa (Database Management System-DBMS) yang tersedia secara *open source* (gratis), *software* ini sangat disukai bahkan sangat populer dikalangan pemrograman web. Software ini dapat memproses data yang berjumlah besar.

*MySQL* merupakan *software* yang bersifat multi user dan juga mampu mengirim serta menerima data dengan sangat cepat.

Ada 3 Komponen MySQL

1. Data Definition Language (DDL)

Data Definition Language berisi perintah-perintah dari SQL yang dapat digunakan untuk mendefinisikan skema dari database. DDL berkaitan dengan deskripsi dari skema database dan digunakan untuk membuat dan memodifikasi struktur dari objek database di dalam database yang ada. DDL terdiri dari beberapa perintah seperti; Create, Drop, Alter, Comment, Rename, dan lain-lain.

2. Data Manipulation Language (DML)

DML atau Data Manipulation Language merupakan perintah dari SQL yang berhubungan dengan manipulasi dan mengambil data dalam database relasional. Data Manipulation Language terdiri dari berbagai contoh perintah seperti; Insert, Update, Delete.

3. Data Control Language (DCL)

Data Control Language berfungsi untuk melakukan pengontrolan data yang banyak berhubungan dengan izin, hak dan kontrol lainnya dalam sebuah sistem database. DCL terdiri dari dua perintah yaitu; Grant dan Revoke.

Perintah Perintah Dasar MySQL

### 1. Create database

Perintah ini adalah untuk membuat database baru. Perintah yang bisa kamu tuliskan adalah sebagai berikut:

```
CREATE DATABASE databasename;
```

### 2. Create table

```
CREATE TABLE table_name (  
column_1 datatype,  
column_2 datatype,  
column_3 datatype  
);
```

### 3. Insert Into

Kemudian, perintah ini dapat digunakan untuk memasukan data baru ke dalam database. Berikut cara penulisannya:

```
INSERT INTO table_name (column_1, column_2, column_3)  
VALUES (value_1, 'value_2', value_3);
```

Kemudian, kamu bisa menuliskan perintah Insert untuk menambahkan baris baru.

### 4. Select

Perintah ini digunakan saat kamu ingin mengambil data dari database. Cara penulisan perintahnya adalah:

```
SELECT column_name  
FROM table_name;
```

### 5. Update

Kamu dapat menggunakannya untuk memperbarui data pada database dan statement update ini artinya memungkinkan kamu untuk mengedit baris di dalam tabel. Berikut cara penulisannya:

```
UPDATE table_name  
SET some_column = some_value  
WHERE some_column = some_value;
```

### 6. Delete



Seperti namanya, perintah ini untuk menghapus data dari database. Berikut cara penulisan perintahnya:

```
DELETE FROM table_name  
WHERE some_column = some_value;
```

#### 7. Alter Table

Perintah ini dapat digunakan jika kamu ingin memodifikasi tabel. Kamu dapat menambahkan, menghapus atau memodifikasi kolom yang ada pada tabel yang sudah ada. Berikut cara penulisannya:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

### 2.2.4 PHP

Menurut Raharjo, Budi, (2009), PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web. Aplikasi web adalah aplikasi yang disimpan dan dieksekusi (oleh PHP Engine) di lingkungan web *server*. Setiap permintaan yang dilakukan oleh user melalui aplikasi akan dikembalikan lagi ke hadapan *user*. Dengan aplikasi web, halaman yang tampil di layar web browser dapat bersifat dinamis, tergantung dari nilai data atau parameter yang dikirimkan oleh user ke web server.

Berikut adalah fungsi fungsi umum yang ada di PHP:

#### 1. Fungsi String

Fungsi string digunakan memanipulasi string untuk berbagai macam kebutuhan. Disini akan dibahas beberapa fungsi string yang sering digunakan dalam membuat program aplikasi web.

#### 2. AddSlashes

Digunakan untuk menambahkan karakter backslash ( \ ) pada suatu string. Hal ini penting digunakan pada *query* string untuk database, misalkan pada *MySQL*. Beberapa karakter yang akan ditambahkan tanda backslash adalah karakter tanda petik satu ( ' ), karakter petik dua ( " ), backslash ( \ ) dan karakter NULL. Sintaks: addslashes(string)

3. StripSlashes Digunakan untuk menghilangkan karakter backslash ( \ ) pada suatu string. Sintaks: string stripslashes(string) *Crypt* Digunakan untuk meng-encrypt dengan metode DES suatu string. Fungsi ini sering digunakan untuk mengacak string password sebelum disimpan dalam database. Dalam penggunaan fungsi *crypt* ini dapat ditambahkan parameter string 'salt'. Parameter 'salt' ini ditambahkan untuk menentukan basis pengacakan. 'Salt' string terdiri atas 2 karakter. Jika 'salt' string tidak ditambahkan pada fungsi *crypt* maka PHP akan menentukan sendiri 'salt' string tersebut secara acak. Sintaks: crypt(string [ , salt ] ) *Echo* Digunakan untuk mencetak isi suatu string atau argumen. Sintaks: echo( string argumen1, string argumen2 , ....) *Explode* Digunakan untuk memecah-mecah suatu string berdasarkan tanda pemisah tertentu dan dimasukkan hasilnya ke dalam suatu variabel array. Sintaks: explode(string pemisah , string [, int limit] )

Contoh:

```
$namahari = "minggu senin selasa rabu kamis jumat
sabtuhari"; $hari = explode(" ", $namahari)
```

Sebelum memanipulasi tabel dalam database *MySQL* (perintah select, insert, delete, update), diperlukan adanya koneksi ke *MySQL* dalam script PHP-nya. Tidak

ada sesuatu yang khusus dalam membuat koneksi dengan MySQL, PHP telah menyediakan fungsi built-in untuk membuat koneksi dengan MySQL. Struktur : `mysql_connect(address, username, password)`; `address` merupakan alamat IP (internet protocol) atau nama host komputer dimana software MySQL server bekerja („localhost“ bisa digunakan, selama web server dan MySQL bekerja dalam satu komputer). `username` merupakan nama user yang berhak untuk mengakses database MySQL (jika tidak diubah, nama user standarnya adalah `root`). `password` merupakan kata sandi yang telah dibuat di database MySQL (secara default, aplikasi wamp5 membiarkan password kosong tidak diisi). Karena fungsi `mysql_connect()` membutuhkan nilai keluaran, maka hasil koneksi harus dilimpahkan pada suatu variabel, sehingga koneksi ke MySQL yang harus dibuat menjadi:

```
<?php
// Koneksi ke server MySQL
$address = 'localhost';
$username = 'root';
$password = '';
$conn     = @mysql_connect($address, $username,
$password);
if (!$conn) {
    echo '<p>Koneksi ke server database gagal dibuat
untuk saat
ini.</p>';
    exit();
}
?>
```

**Gambar 2. 2 Kode Program Koneksi ke server MySQL**

Symbol @, lebih dikenal dengan error suppression operator, berfungsi untuk memberitahukan fungsi php, jika ada kesalahan maka akan ditampilkan pesan dengan bahasa yang mudah dipahami oleh programmer. Setelah koneksi ke MySQL terbentuk, maka langkah selanjutnya adalah memilih database mana yang akan

diaktifkan untuk dilakukan operasi manipulasi. Struktur :  
 mysql\_select\_db(\$dbname, \$conn\_var); Dengan anggapan bahwa kita telah  
 membuat database dengan nama “databaseku”, maka script yang harus dibuat  
 adalah:

```

<?php
// Koneksi ke server MySQL
$address = 'localhost';
$username = 'root';
$password = '';
$conn     = @mysql_connect($address, $username,
$password);
if (!$conn) {
    echo '<p>Koneksi ke server database gagal dibuat
    untuk saat
    ini.</p>';
    exit();
}
// Memilih database dan mengaktifkannya
$conn_var = 'DatabaseKu';
if (!@mysql_select_db($conn_var)) {
    exit('<p>Pencarian database DatabaseKu tidak
    ditemukan. </p>');
}
?>

```

**Gambar 2. 3 Contoh Kode Program Koneksi ke Database**

### 2.2.5 Service Worker

Progressive Web App harus cepat, yang berarti bahwa tetap berfungsi saat online, offline, dan pada koneksi yang lambat serta tidak stabil. Untuk mencapainya, PWA harus menyimpan sementara tampilan antarmuka dari aplikasi website dengan menggunakan *service worker*, sehingga selalu tersedia dengan cepat. *Service worker* adalah *skrip* yang dijalankan browser yang berjalan di latar belakang yang terpisah dari laman web, yang membuka pintu ke berbagai fitur dimana tidak memerlukan laman web ataupun interaksi pengguna.

Saat ini *service worker* sudah menyertakan berbagai fitur seperti pemberitahuan, push, dan sinkronisasi latar belakang. *Service worker* juga berkemampuan mencegah dan menangani permintaan jaringan termasuk mengelola cache respons melalui program. Yang membuat API ini menarik adalah karena memungkinkan pengguna mendukung pengalaman offline. Sebelum *service worker*, ada satu API lain yang memberikan pengguna pengalaman offline di website, yaitu *AppCache*.

Namun pada *AppCache* jumlah gotcha yang ada serta fakta bahwa meskipun desain bekerja dengan sangat baik, untuk laman aplikasi web tunggal, namun ternyata tidak begitu baik untuk situs multi-laman. *Service Worker* telah didesain untuk menghindari titik-titik menyulitkan yang sudah umum tersebut. Untuk memasang *service worker*, perlu mendaftarkannya sebagai file *JavaScript*. Mendaftarkan *service worker* akan menyebabkan browser memulai 10 langkah pemasangan *service worker* di latar belakang. Biasanya selama langkah pemasangan, perlu menyimpan sementara beberapa asset statis. Jika semua file berhasil disimpan sementara, maka *service worker* akan terpasang. Jika ada file yang gagal diunduh dan disimpan sementara, maka langkah pemasangan akan gagal dan *service worker* tidak akan diaktifkan sehingga membuat aplikasi website tersebut tidak dapat berjalan secara offline. Setelah *service worker* aktif, maka secara bersamaan akan menangani manajemen penyimpanan sementara yang sebelumnya. Lalu *service worker* akan mengontrol semua halaman yang berada dalam cakupannya, walaupun laman yang mendaftarkan *service worker* untuk pertama kali tidak akan di control hingga dimuat ulang. jika *service worker* sudah

terkontrol, *service worker* akan berada dalam salah satu dari dua keadaan, yaitu :  
*service worker* akan dihentikan untuk menghemat memori, atau *service worker*  
akan menangani pesan untuk menjalankan permintaan saat jaringan buruk.